Docket No. 216446US2SRD/btm

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Shigehiro ASANO, et al.
GAU: 2185
SERIAL NO: 09/989,028
EXAMINER:
FILED: November 21, 2001
FOR: MULTIPROCESSOR SYSTEM AND CONTROL METHOD THEREOF

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number [US App No], filed [US App Dt], is claimed pursuant to the provisions of **35 U.S.C. §120**.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number , filed , is claimed pursuant to the provisions of **35 U.S.C. §119(e)**.
- ■ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of **35 U.S.C. §119**, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

COUNTRY	APPLICATION NUMBER	MONTH/DAY/YEAR
JAPAN	2000-356237	November 22, 2000

Certified copies of the corresponding Convention Application(s)

- ■ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .
  Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
  (B) Application Serial No.(s)
  - ☐ are submitted herewith
  - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Joseph A. Scafetta Jr.

Marvin J. Spivak
Registration No. 24,913
Joseph A. Scafetta, Jr.
Registration No. 26,803

22850
Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/98)

09/989,028

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２０００年１１月２２日

出願番号
Application Number: 特願２０００－３５６２３７

出願人
Applicant(s): 株式会社東芝

２００１年１２月２１日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号　出証特２００１－３１１０９１５

【書類名】　特許願

【整理番号】　A000006367

【提出日】　平成12年11月22日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 15/00

【発明の名称】　マルチプロセッサシステムおよびその制御方法

【請求項の数】　7

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

【氏名】　浅野　滋博

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝マイクロエレクトロニクスセンター内

【氏名】　斎藤　光男

【特許出願人】

【識別番号】　000003078

【氏名又は名称】　株式会社　東芝

【代理人】

【識別番号】　100058479

【弁理士】

【氏名又は名称】　鈴江　武彦

【電話番号】　03-3502-3181

【選任した代理人】

【識別番号】　100084618

【弁理士】

【氏名又は名称】　村松　貞男

【選任した代理人】

【識別番号】　100068814

【弁理士】

【氏名又は名称】　坪井　淳

【選任した代理人】

【識別番号】　100092196

【弁理士】

【氏名又は名称】　橋本　良郎

【選任した代理人】

【識別番号】　100091351

【弁理士】

【氏名又は名称】　河野　哲

【選任した代理人】

【識別番号】　100088683

【弁理士】

【氏名又は名称】　中村　誠

【選任した代理人】

【識別番号】　100070437

【弁理士】

【氏名又は名称】　河井　将次

【手数料の表示】

【予納台帳番号】　011567

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　マルチプロセッサシステムおよびその制御方法

【特許請求の範囲】

【請求項１】　マスタプロセッサと、ローカルメモリを各々が有し前記マスタプロセッサからのコマンドに従ってそれぞれ制御される複数のプロセッサエレメントと、前記複数のプロセッサエレメントに共有されるグローバルメモリとを具備するマルチプロセッサシステムにおいて、

前記マスタプロセッサからのコマンドに従って、前記グローバルメモリと前記各プロセッサエレメントのローカルメモリとの間のＤＭＡ転送を行うＤＭＡ実行手段と、

前記マスタプロセッサが各コマンドに対する応答を待つことなく次のコマンドを発行できるように前記ＤＭＡ実行手段および前記各プロセッサエレメントにそれぞれ対応して設けられたコマンド蓄積用バッファと、

発行済みで応答が返ってきてないコマンドの数を管理し、すべての発行済みのコマンドに対して応答が返ってきたときにそれを前記マスタプロセッサに通知する未応答コマンド数管理手段とを具備することを特徴とするマルチプロセッサシステム。

【請求項２】　前記マスタプロセッサ上で並列実行される複数のスレッドそれぞれに固有の識別子を用いて、前記各コマンドをその発行元のスレッドの識別子と共に前記マスタプロセッサから前記ＤＭＡ実行手段または前記各プロセッサエレメントに発行する手段と、

前記発行されたコマンドに対応する応答を該当するスレッドの識別子と共に前記ＤＭＡ実行手段または前記各プロセッサエレメントから前記マスタプロセッサに発行する手段とをさらに具備し、

前記未応答コマンド数管理手段は発行済みで応答が返ってきてないコマンドの数を前記スレッド毎に個々に管理し、すべての発行済みのコマンドに対して応答が返ってきたスレッドに対してコマンド処理の完了を通知するように構成されていることを特徴とする請求項１記載のマルチプロセッサシステム。

【請求項３】　マスタプロセッサと、ローカルメモリを各々が有し前記マス

タプロセッサからのコマンドに従ってそれぞれ制御される複数のプロセッサエレメントと、前記複数のプロセッサエレメントに共有されるグローバルメモリとを具備するマルチプロセッサシステムにおいて、

前記マスタプロセッサからのコマンドに従って、前記グローバルメモリと前記各プロセッサエレメントのローカルメモリとの間のＤＭＡ転送を行うＤＭＡ実行手段と、

前記マスタプロセッサが各コマンドに対する応答を待つことなく次のコマンドを発行できるように前記ＤＭＡ実行手段および前記各プロセッサエレメントにそれぞれ対応して設けられたコマンド蓄積用バッファと、

発行済みで応答が返ってきてないコマンドの数を管理し、すべての発行済みのコマンドに対して応答が返ってきたときにそれを前記マスタプロセッサに通知する手段と、

前記マスタプロセッサ上で並列実行される複数のスレッドそれぞれに固有の識別子を用いて、前記各コマンドをその発行元のスレッドの識別子と共に前記マスタプロセッサから前記ＤＭＡ実行手段または前記各プロセッサエレメントに発行する手段と、

前記発行されたコマンドに対応する応答を該当するスレッドの識別子と共に前記ＤＭＡ実行手段または前記各プロセッサエレメントから前記マスタプロセッサに発行する手段と、

前記複数のスレッドそれぞれの識別子を用いて、前記ローカルメモリの各記憶領域毎にどのスレッドに対応するプログラムまたはデータが格納されているかを管理する管理テーブルとを具備し、

前記ＤＭＡ実行手段および前記各プロセッサエレメントは、それぞれ対応するコマンド蓄積用バッファに蓄積されている前記各スレッドからのコマンドの中で、実行可能なコマンドを前記管理テーブルを参照して特定し、その特定したコマンドを実行することを特徴とするマルチプロセッサシステム。

【請求項４】　前記各プロセッサエレメントは、前記管理テーブルを参照することにより、対応するコマンド蓄積用バッファに蓄積されている各スレッドからのコマンドの中で、ローカルメモリ内に該当するスレッドの処理対象データが

用意されたコマンドから順に処理を開始することを特徴とする請求項３記載のマルチプロセッサシステム。

【請求項５】　前記管理テーブルは、前記ローカルメモリ内に前記プロセッサエレメントのコマンド処理で得られた処理結果のデータが用意されたか否かをスレッド毎に示す情報を含み、

前記ＤＭＡ実行手段は、前記管理テーブルを参照することにより、コマンド蓄積用バッファに蓄積されている前記各スレッドからのコマンドの中で、前記ローカルメモリ内に該当するスレッドの処理結果のデータが用意されたコマンドから順に処理を開始して、前記ローカルメモリから前記グローバルメモリへのＤＭＡ転送を行うことを特徴とする請求項４記載のマルチプロセッサシステム。

【請求項６】　マスタプロセッサと、ローカルメモリを各々が有し前記マスタプロセッサからのコマンドに従ってそれぞれ制御される複数のプロセッサエレメントと、前記複数のプロセッサエレメントに共有されるグローバルメモリとを具備するマルチプロセッサシステムの動作を制御するための制御方法であって、

前記マルチプロセッサシステムには、前記マスタプロセッサからのコマンドに従って、前記グローバルメモリと前記各プロセッサエレメントのローカルメモリとの間のＤＭＡ転送を行うＤＭＡ実行手段、及び前記ＤＭＡ実行手段および前記各プロセッサエレメントにはそれぞれ前記マスタプロセッサが各コマンドに対する応答を待つことなく次のコマンドを発行できるように複数のコマンドを蓄積可能なコマンド蓄積用バッファが設けられており、

前記マスタプロセッサから前記ＤＭＡ実行手段および前記各プロセッサエレメントに対して、各コマンドに対する応答を待つことなく次のコマンドを発行するステップと、

発行済みで応答が返ってきてないコマンドの数を管理し、すべての発行済みのコマンドに対して応答が返ってきたときにそれを前記マスタプロセッサに通知するステップとを具備することを特徴とするマルチプロセッサシステムの制御方法。

【請求項７】　マスタプロセッサと、ローカルメモリを各々が有し前記マスタプロセッサからのコマンドに従ってそれぞれ制御される複数のプロセッサエレ

メントと、前記複数のプロセッサエレメントに共有されるグローバルメモリとを具備するマルチプロセッサシステムの動作を制御するための制御方法であって、

前記マルチプロセッサシステムには、前記マスタプロセッサからのコマンドに従って、前記グローバルメモリと前記各プロセッサエレメントのローカルメモリとの間のＤＭＡ転送を行うＤＭＡ実行手段、及び前記ＤＭＡ実行手段および前記各プロセッサエレメントにはそれぞれ前記マスタプロセッサが各コマンドに対する応答を待つことなく次のコマンドを発行できるように複数のコマンドを蓄積可能なコマンド蓄積用バッファが設けられており、

発行済みで応答が返ってきてないコマンドの数を管理し、すべての発行済みのコマンドに対して応答が返ってきたときにそれを前記マスタプロセッサに通知するステップと、

前記マスタプロセッサ上で並列実行される複数のスレッドそれぞれに固有の識別子を用いて、前記各コマンドをその発行元のスレッドの識別子と共に前記マスタプロセッサから前記ＤＭＡ実行手段または前記各プロセッサエレメントに発行するステップと、

前記発行されたコマンドに対応する応答を該当するスレッドの識別子と共に前記ＤＭＡ実行手段または前記各プロセッサエレメントから前記マスタプロセッサに発行するステップと、

前記複数のスレッドそれぞれの識別子を用いて、前記ローカルメモリの使用領域を管理テーブルによってスレッド毎に管理するステップとを具備し、

前記ＤＭＡ実行手段および前記各プロセッサエレメントは、それぞれ対応するコマンド蓄積用バッファに蓄積されている前記各スレッドからのコマンドの中で、実行可能なコマンドを前記管理テーブルを参照して特定し、その特定したコマンドを実行することを特徴とするマルチプロセッサシステムの制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はマルチプロセッサシステムおよびその制御方法に関し、特にグローバルメモリと各プロセッサエレメントのローカルメモリとの間でデータ／プログラ

ムの転送をＤＭＡ転送によって行うマルチプロセッサシステムおよびその制御方法に関する。

【０００２】

【従来の技術】

従来より、マルチプロセッサ方式は計算機を高速化する手法として広く知られている。マルチプロセッサ方式には、メモリを共有する共有メモリ方式（密結合マルチプロセッサ方式）と、メモリをプロセッサに分散して持つ疎結合マルチプロセッサ方式とがある。

【０００３】

共有メモリ方式では共有メモリを通してプロセッサ間の通信を行うことができるのでプログラミングが簡単な反面、データの整合性を保持した状態で各プロセッサが共有メモリを参照できるようにするための共有メモリアクセスに関する特別な仕組みが必要となり、ハードウェアが複雑になるという欠点がある。また、疎結合マルチプロセッサ方式では共有メモリを用いたプロセッサ間通信は行われないのでハードウェアはその分簡単化できるが、プロセッサ間通信のための機能をプロセッサ毎にそれを制御するプログラムによって提供する必要があり、プログラム作成が困難であるという問題がある。

【０００４】

そこで、疎結合マルチプロセッサ方式の制御を容易に行うために、各プロセッサの制御を行うためのマスタプロセッサを設け、そのマスタプロセッサが他の各プロセッサエレメントにコマンドを送る方法が知られている。マスタプロセッサ上で他の各プロセッサエレメントに送るコマンドの順序およびタイミングを制御することで、マルチプロセッサシステム全体の動作を容易に制御することが可能となる。

【０００５】

【発明が解決しようとする課題】

ところで、疎結合マルチプロセッサ方式では各プロセッサエレメントがローカルなメモリを持つが、疎結合マルチプロセッサ方式であっても各プロセッサエレメントで共通に使用可能なメモリ（グローバルメモリ）が存在した方がプログラ

ミングは容易であり、また各ローカルメモリのメモリサイズの縮小化を図ることもできる。

【０００６】

しかし、一般にローカルメモリに比べ、グローバルメモリに対するアクセスにはバス調停やその他の要因で多くの時間を要する。メモリアクセスのためにプロセッサ資源が長い間専有されてしまうことはスループットの低下をもたらすことになる。

【０００７】

これを改善するため、最近では、グローバルメモリと各プロセッサエレメントのローカルメモリとの間のデータ／プログラムの転送をＤＭＡコントローラによって行う仕組みが考えられている。

【０００８】

この場合、マスタプロセッサで実行されるプログラムは各プロセッサエレメントおよびＤＭＡコントローラを制御する手順を記述したものになる。このプログラムをマルチスレッドで記述することにより、複数のプロセッサエレメントを効率よく使用することができる。

【０００９】

ところが、個々のプロセッサエレメントの処理時間およびＤＭＡにかかる時間は既知ではないので、マスタプロセッサ上で実行されるマルチスレッドプログラムで各プロセッサエレメントおよびＤＭＡコントローラを制御する場合でも、各スレッドに対応する処理動作を該当するプロセッサエレメントに無駄無く割り当てるのは実際上困難である。空いているプロセッサエレメントの時間を減らすためには、次の二つの問題点を解決しなければならない。

【００１０】

第１の問題点は、ＤＭＡとプロセッサエレメントの処理には依存性があるので、この依存性を制御するためにマスタプロセッサを用いるが、ＤＭＡおよびプロセッサエレメントの処理の終了の度にマスタプロセッサへ割り込みやスレッドの切り替えを起こしたのではマスタプロセッサの動作が非効率的であるということである。特に、マスタプロセッサによって制御すべきプロセッサエレメントの数

が増えると、マスタプロセッサへ割り込みやスレッドの切り替えの処理が頻繁に発生し、処理効率が低下することになる。

【００１１】

第２の問題点は、マスタプロセッサ上で実行される２以上のスレッドに対応する処理動作をあるプロセッサエレメントに実行させる場合、あるスレッドの制御でローカルメモリにＤＭＡ転送したデータを、別のスレッドに対応する処理で使ってしまうということが起こり得る点である。

【００１２】

例えば、スレッドＡとスレッドＢがマスタプロセッサ上で並列実行されており、それらスレッドそれぞれの制御でプロセッサエレメントがローカルメモリ上のデータを処理する場合を考える。この場合、スレッドＡとスレッドＢの切り換えのタイミングやＤＭＡおよびプロセッサエレメントの処理時間との関係によっては、グローバルメモリからローカルメモリにＤＭＡ転送されたスレッドＡ用のデータがスレッドＡに対応するプロセッサエレメントの処理動作で実際に処理される前に、スレッドＡからスレッドＢへの切り換えが行われ、プロセッサエレメントのスレッドＢに対応する処理動作でスレッドＡ用のデータが使われてしまうことがある。これにより、扱うデータに不整合が生じる。

【００１３】

本発明は上述の事情を考慮してなされたものであり、マスタプロセッサの負荷の増大や、扱うデータの不整合に関する問題を招くことなく、空いているプロセッサエレメントの時間を減らしてスループットの向上を図ることが可能なマルチプロセッサシステムおよびその制御方法を提供することを目的とする。

【００１４】

【課題を解決するための手段】

上述の課題を解決するため、本発明は、マスタプロセッサと、ローカルメモリを各々が有し前記マスタプロセッサからのコマンドに従ってそれぞれ制御される複数のプロセッサエレメントと、前記複数のプロセッサエレメントに共有されるグローバルメモリとを具備するマルチプロセッサシステムにおいて、前記マスタプロセッサからのコマンドに従って、前記グローバルメモリと前記各プロセッサ

エレメントのローカルメモリとの間のＤＭＡ転送を行うＤＭＡ実行手段と、前記マスタプロセッサが各コマンドに対する応答を待つことなく次のコマンドを発行できるように前記ＤＭＡ実行手段および前記各プロセッサエレメントにそれぞれ対応して設けられたコマンド蓄積用バッファと、発行済みで応答が返ってきてないコマンドの数を管理し、すべての発行済みのコマンドに対して応答が返ってきたときにそれを前記マスタプロセッサに通知する未応答コマンド数管理手段とを具備することを特徴とする。

【００１５】

この発明によれば、ＤＭＡ実行手段および各プロセッサエレメントにそれぞれコマンド蓄積用バッファが設けられているので、マスタプロセッサからＤＭＡ実行手段や各プロセッサエレメントへのコマンドは複数まとめて発行することができ、先に送ったコマンドに対する応答を待たずに次のコマンドを発行することができる。さらに、発行済みで応答が返ってきてないコマンドの数の管理が行われ、すべての発行済みのコマンドに対して応答が返ってきたときにそれがマスタプロセッサに通知される。この通知により、はじめてマスタプロセッサにスレッド切り替え等の動作をとらせれば、ＤＭＡと各プロセッサエレメントの処理の依存性を考慮しつつ、マスタプロセッサの負荷を低減することができる。よって、個々のコマンド発行およびそれに対する応答の度にスレッドの切り換えが行われるシステムに比べ、システム全体のスループットを向上させることができる。もちろん、マスタプロセッサ上のプログラムがシングルスレッドで動作している場合であっても、割り込み処理等の回数を減らすことが可能となり、マスタプロセッサの負荷を低減することができる。

【００１６】

また、スレッド毎に固有の識別子を設け、その識別子と共にコマンドを発行し、コマンドに対する応答も識別子と共に返すようにすることが好ましい。そして、発行済みで応答が返ってきてないコマンドの数をスレッド毎に個々に管理し、すべての発行済みのコマンドに対して応答が返ってきたスレッドに対してコマンド処理の完了を通知することにより、スレッド間の切り換えをよりスムーズに行うことが可能となる。

【００１７】

また、本発明は、マスタプロセッサと、ローカルメモリを各々が有し前記マスタプロセッサからのコマンドに従ってそれぞれ制御される複数のプロセッサエレメントと、前記複数のプロセッサエレメントに共有されるグローバルメモリとを具備するマルチプロセッサシステムにおいて、前記マスタプロセッサからのコマンドに従って、前記グローバルメモリと前記各プロセッサエレメントのローカルメモリとの間のＤＭＡ転送を行うＤＭＡ実行手段と、前記マスタプロセッサが各コマンドに対する応答を待つことなく次のコマンドを発行できるように前記ＤＭＡ実行手段および前記各プロセッサエレメントにそれぞれ対応して設けられたコマンド蓄積用バッファと、発行済みで応答が返ってきてないコマンドの数を管理し、すべての発行済みのコマンドに対して応答が返ってきたときにそれを前記マスタプロセッサに通知する手段と、前記マスタプロセッサ上で並列実行される複数のスレッドそれぞれに固有の識別子を用いて、前記各コマンドをその発行元のスレッドの識別子と共に前記マスタプロセッサから前記ＤＭＡ実行手段または前記各プロセッサエレメントに発行する手段と、前記発行されたコマンドに対応する応答を該当するスレッドの識別子と共に前記ＤＭＡ実行手段または前記各プロセッサエレメントから前記マスタプロセッサに発行する手段と、前記複数のスレッドそれぞれの識別子を用いて、前記ローカルメモリの各記憶領域毎にどのスレッドに対応するプログラムまたはデータが格納されているかを管理する管理テーブルとを具備し、前記ＤＭＡ実行手段および前記各プロセッサエレメントは、それぞれ対応するコマンド蓄積用バッファに蓄積されている前記各スレッドからのコマンドの中で、実行可能なコマンドを前記管理テーブルを参照して特定し、その特定したコマンドを実行することを特徴とする。

【００１８】

この構成によれば、ローカルメモリの各記憶領域毎にどのスレッドに対応するプログラムまたはデータが格納されているかを管理する管理テーブルが設けられており、上述の効果に加え、扱うデータの不整合に関する問題を解消することができる。さらに、ＤＭＡ実行手段および各プロセッサエレメントは、それぞれ対応するコマンド蓄積用バッファに蓄積されている各スレッドからのコマンドの中

で、実行可能なコマンドを管理テーブルを参照して特定できるので、実行可能なコマンドから順に正しく実行することが可能となり、ＤＭＡ実行手段および各プロセッサエレメントの空き時間を減らすことが可能となる。

【００１９】

【発明の実施の形態】

以下、図面を参照して本発明の実施形態を説明する。

［実施形態１］

図１には、本発明の第１実施形態に係るマルチプロセッサシステムの構成が示されている。このマルチプロセッサシステムは疎結合方式のマルチプロセッサシステムであり、図示のように、バス１０、マスタプロセッサ（ＭＰ）１１、および２つのプロセッサエレメント（ＰＥ）１２－１，１２－２を備えている。マスタプロセッサ（ＭＰ）１１は２つのプロセッサエレメント（ＰＥ）１２－１，１２－２を制御するためのものであり、プロセッサエレメント（ＰＥ）１２－１，１２－２による処理動作は全てマスタプロセッサ（ＭＰ）１１から発行されるコマンドに従って実行される。

【００２０】

２つのプロセッサエレメント（ＰＥ）１２－１，１２－２の各々はそれぞれ処理対象のプログラムおよびデータを格納するためのローカルメモリを持つ。プログラムメモリ（ＰＬＳ：Ｐｒｏｇｒａｍ　Ｌｏｃａｌ　Ｓｔｏｒａｇｅ）１３－１およびデータローカルメモリ（ＤＬＳ：Ｄａｔａ　Ｌｏｃａｌ　Ｓｔｏｒａｇｅ）１４－１はプロセッサエレメント（ＰＥ）１２－１のローカルメモリであり、プログラムローカルメモリ（ＰＬＳ）１３－１にはプロセッサエレメント（ＰＥ）１２－１に実行させるべきプログラムが格納され、またデータローカルメモリ（ＤＬＳ）１４－１にはプロセッサエレメント（ＰＥ）１２－１によって処理させるべきデータ等が格納される。同様に、プログラムメモリ（ＰＬＳ：Ｐｒｏｇｒａｍ　Ｌｏｃａｌ　Ｓｔｏｒａｇｅ）１３－２およびデータローカルメモリ（ＤＬＳ：Ｄａｔａ　Ｌｏｃａｌ　Ｓｔｏｒａｇｅ）１４－２はプロセッサエレメント（ＰＥ）１２－２のローカルメモリであり、プログラムローカルメモリ（ＰＬＳ）１３－２にはプロセッサエレメント（ＰＥ）１２－２に実行させるべき

プログラムが格納され、またデータローカルメモリ（ＤＬＳ）１４－２にはプロセッサエレメント（ＰＥ）１２－２によって処理させるべきデータ等が格納される。

【００２１】

さらに、バス１０にはＤＭＡコントローラ１５－１，１５－２が接続されると共に、メモリコントローラ１６を介してグローバルメモリ（ＧＭ）１７が接続されている。グローバルメモリ（ＧＭ）１７はプロセッサエレメント（ＰＥ）１２－１，１２－２が共通に使用可能な一種の共有メモリであり、ここにはマスタプロセッサ（ＭＰ）１１によって実行されるプログラムおよび処理データが格納されるほか、プロセッサエレメント（ＰＥ）１２－１，１２－２によって実行されるプログラムおよび処理データも格納される。このグローバルメモリ（ＧＭ）１７から必要なプログラムをプログラムローカルメモリ（ＰＬＳ）１３－１，１３－２にそれぞれ転送することにより、プログラムローカルメモリ（ＰＬＳ）１３－１，１３－２のメモリサイズを大きくせずに、比較的大きなプログラムでもそれをプロセッサエレメント（ＰＥ）１２－１，１２－２にそれぞれ実行させることができる。

【００２２】

また、グローバルメモリ（ＧＭ）１７とデータローカルメモリ（ＤＬＳ）１４－１，１４－２それぞれとの間のデータ転送により、処理対象のデータをグローバルメモリ（ＧＭ）１７からデータローカルメモリ（ＤＬＳ）１４－１，１４－２に転送してプロセッサエレメント（ＰＥ）１２－１，１２－２にそれぞれ処理させたり、それぞれの処理結果のデータをデータローカルメモリ（ＤＬＳ）１４－１，１４－２からグローバルメモリ（ＧＭ）１７に転送するなどの制御を容易に行うことができる。

【００２３】

ＤＭＡコントローラ１５－１，１５－２は、それぞれマスタプロセッサ（ＭＰ）１１からのコマンドに従って、対応するローカルメモリとグローバルメモリ（ＧＭ）１７との間でデータ／プログラムのＤＭＡ転送を実行する。これらＤＭＡコントローラ１５－１，１５－２の存在により、プロセッサエレメント１２－１

，１２－２およびマスタプロセッサ（ＭＰ）１１の資源を専有することなく、ローカルメモリとグローバルメモリ（ＧＭ）１７との間のデータ／プログラムの転送を効率よく行うことができる。

【００２４】

上述したように、プロセッサエレメント１２－１，１２－２の処理およびＤＭＡコントローラ１５－１，１５－２の動作は全てマスタプロセッサ（ＭＰ）１１によって制御される。この制御処理は、マスタプロセッサ（ＭＰ）１１上で実行されるマルチスレッドプログラムによって実行される。マルチスレッドプログラムを構成する複数のスレッドはマスタプロセッサ（ＭＰ）１１上で並列に実行され、それらスレッド毎に必要な演算処理等がプロセッサエレメント１２－１または１２－２を用いて実行される。つまり、各スレッドはＤＭＡコントローラ１５－１または１５－２、およびプロセッサエレメント１２－１または１２－２に送るコマンドの順序およびタイミングを制御することで、必要な演算処理をプロセッサエレメントに実行させ、その処理結果をグローバルメモリ（ＧＭ）１７上に転送させることができる。基本的には、以下の手順で処理が行われる。

【００２５】

１）グローバルメモリ（ＧＭ）からプログラムローカルメモリ（ＰＬＳ）へのプログラムのＤＭＡ転送

２）グローバルメモリ（ＧＭ）からデータローカルメモリ（ＤＬＳ）への処理対象データのＤＭＡ転送

３）プロセッサエレメントによる処理

４）データローカルメモリ（ＤＬＳ）からグローバルメモリ（ＧＭ）への処理結果データの転送

本マルチプロセッサシステムは例えば産業用または家庭用機器への組み込みコンピュータとして使用される。この場合、図１に示した複数の機能モジュールのうち、グローバルメモリ（ＧＭ）１７を除く他の全てのモジュールは１チップＬＳＩ上に集積形成することができる。

【００２６】

さらに、本マルチプロセッサシステムでは、一連の制御に必要な幾つかのコマ

ンドをまとめて扱えるようにするために、プロセッサエレメント（ＰＥ）１２－１，１２－２にはそれぞれ複数のコマンドを蓄積可能なコマンドプーリングバッファ１２１が設けられ、またＤＭＡコントローラ１５－１，１５－２にもそれぞれ複数のコマンドを蓄積可能なコマンドプーリングバッファ１３１が設けられている。これらコマンドプーリングバッファ１２１，１３１の各々は追い越し制御可能な一種のコマンドキュー（プール）である。

【００２７】

これらコマンドプーリングバッファ１２１，１３１の存在により、マスタプロセッサ（ＭＰ）１１からＤＭＡコントローラ１５－１，１５－２や各プロセッサエレメント（ＰＥ）へのコマンドは複数まとめて発行することができ、先に送ったコマンドに対する応答を待たずに次のコマンドを発行することができる。さらに、発行済みで応答が返ってきてないコマンドの数の管理はマスタプロセッサ（ＭＰ）１１に設けられたカウンタアレイ１１１によって行われ、すべての発行済みのコマンドに対して応答が返ってきたときにそのことがマスタプロセッサ（ＭＰ）１１に通知される。この通知により、はじめてマスタプロセッサ（ＭＰ）１１にスレッド切り替え等の動作をとらせることにより、ＤＭＡと各プロセッサエレメント１２－１，１２－２の処理の依存性を考慮しつつ、マスタプロセッサ（ＭＰ）１１の負荷を低減することができる。この場合、実際には、スレッド間の切り換えをよりスムーズに行うために、発行済みで応答が返ってきてないコマンドの数の管理はスレッド毎に行うことが好ましい。

【００２８】

すなわち、本実施形態では、スレッドごとに固有の識別子ＶＴＩＤ（Ｖｉｒｔｕａｌ　Ｔｈｒｅａｄ　ＩＤ）を付ける。さらに、カウンタアレイ１１１にはＶＴＩＤごとに固有のカウンタを設け、マスタプロセッサ（ＭＰ）１１からのコマンドの発行で該当するカウンタをインクリメントする。ＤＭＡコントローラ１５－１，１５－２やプロセッサエレメント（ＰＥ）１２－１，１２－２へはコマンドと共にＶＴＩＤを送出するようにし、ＤＭＡコントローラ１５－１，１５－２やプロセッサエレメント（ＰＥ）１２－１，１２－２からの応答には対応したＶＴＩＤを共に返すようにする。マスタプロセッサ（ＭＰ）１１では応答と共に返

ってきたＶＴＩＤに対応したカウンタをデクリメントする。カウンタが０になれば、はじめて該当するスレッドにそれを通知してスリープ状態から実行状態に起こすことにより、実行スレッドの切り換えをよりスムーズに行うことが可能となる。図２にこの様子を示す。

【００２９】

図２（ａ）は従来の場合に相当するものであり、マスタプロセッサ（ＭＰ）１１上で実行されているスレッドはプロセッサエレメント（ＰＥ）へ２回コマンドを発行したことで２回の応答を受けている。この例ではコマンドを発行したときにスレッドをスリープさせ、応答があったときコマンドを再び起こしているが、これは割り込みであったり、スリープの代わりにビジーウェイトであっても構わない。

【００３０】

図２（ｂ）は本実施形態の場合であり、マスタプロセッサ（ＭＰ）１１上で実行されているスレッドからプロセッサエレメント（ＰＥ）に最初にコマンドをなげた時点でカウンタの値が１になる。その後、２個目のコマンドをプロセッサエレメント（ＰＥ）に投げてこのスレッドをスリープさせる。この時、カウンタの値は２になっている。その後、プロセッサエレメント（ＰＥ）より最初のコマンドに対する応答が返ってきてカウンタの値は１になる。最後にプロセッサエレメント（ＰＥ）より２個目のコマンドに対応する応答が返ってきてカウンタの値が０になり、スリープしていたスレッドを起こす。

【００３１】

あるスレッドがスリープしている間は他のスレッドがマスタプロセッサ（ＭＰ）１１上で実行されるので、一連の処理に必要な幾つかのコマンド処理待ちの間にマスタプロセッサ（ＭＰ）１１は他の処理を行うことができる。コマンドをいくつ発生してからスリープするかについては各スレッドの記述によって決まる。

【００３２】

以上のようにカウンタという簡単なハードウェアを設けることにより、マスタプロセッサ（ＭＰ）１１上のスレッドをスリープさせたり再び起こしたりするオーバーヘッドが削減される。もちろん、割り込みや、ビジーウェイトなどを使っ

た場合も同様にオーバーヘッドは削減される。

【００３３】

このような仕組みを実現するためのハードウェアは以下の図３のような構成が考えられる。

【００３４】

図３ではマスタプロセッサ（ＭＰ）１１とそれにつながるバス１０を示している。マスタプロセッサ（ＭＰ）１１は図示のようにプロセッサモジュール２０１、バスコントローラ２０２、およびカウンタアレイ２０３を備えている。このカウンタアレイ２０３は図１のカウンタアレイ１１１と同じものである。

【００３５】

プロセッサモジュール２０１からコマンドを発行するとき、コマンドは発行元のスレッドのＶＴＩＤと共にバスコントローラ２０２に送られる。また、プロセッサモジュール２０１からカウンタアレイ２０３にはＶＴＩＤが与えられる。カウンタアレイ２０３ではＶＴＩＤに対応したカウンタをインクリメントする。一方、バス１０を介してＶＴＩＤとともに応答が返ってくると、カウンタアレイ２０３の対応するカウンタをデクリメントする。カウンタが０になった場合はプロセッサモジュール２０１への通知が行われ、該当するＶＴＩＤのスレッドが起こされ、すべての発行済みのコマンドに対して応答が返ってきたスレッドに対してコマンド処理の完了が通知される。

【００３６】

次に、図４および図５のフローチャートを参照して、マスタプロセッサ（ＭＰ）１１上で実行される各スレッドの動作と、各スレッドからのコマンドで制御されるＤＭＡコントローラおよびプロセッサエレメントの動作について説明する。

【００３７】

図４は、マスタプロセッサ（ＭＰ）１１上で実行される各スレッドの動作を示している。まず、スレッドはＤＭＡコントローラ１５－１，１５－２やプロセッサエレメント（ＰＥ）１２－１，１２－２に対して、ＶＴＩＤとコマンドの組をＮ個順次発行する（ステップＳ１０１）。この後、スレッドは、対応するＶＴＩＤのカウンタをＮにセットした後（ステップＳ１０２）、スリープ状態に移行さ

れる（ステップＳ１０３）。そして、カウンタアレイ２０３から通知が来ると、カウンタ値が０になったカウンタに対応するＶＴＩＤのスレッドがスリープ状態から起こされ（ステップＳ１０４）、その起こされたスレッドによる処理が続行される。

【００３８】

図５は、ＤＭＡコントローラ１５－１，１５－２およびプロセッサエレメント（ＰＥ）１２－１，１２－２の動作を示している。

【００３９】

ＤＭＡコントローラまたはプロセッサエレメントでマスタプロセッサ（ＭＰ）１１からのコマンドが受け付けられると（ステップＳ１１１）、まずそのコマンドに対応するＶＴＩＤがセーブされた後（ステップＳ１１２）、コマンド処理が実行される（ステップＳ１１３）。このコマンド処理では、ＤＭＡコントローラ１５－１，１５－２の場合にはコマンドで指定されたＤＭＡ転送が実行され、プロセッサエレメントの場合にはコマンドで指定されたプログラム実行処理が行われることになる。

【００４０】

コマンド処理が終了すると、そのコマンドを実行したＤＭＡコントローラまたはプロセッサエレメントは、そのコマンドに対する応答として、セーブしておいたＶＴＩＤと共にコマンド完了通知をマスタプロセッサ（ＭＰ）１１に発行する（ステップＳ１１４）。

【００４１】

以上のように、本マルチプロセッサシステムによれば、スレッドごとの識別子ＶＴＩＤとマスタプロセッサ（ＭＰ）１１に設けたＶＴＩＤごとのカウンタによって発行済みで応答のないコマンド数をスレッドごとに管理し、発行済みのコマンド処理がすべて終わったときにマスタプロセッサ（ＭＰ）１１にそれを通知してスレッド間切り換えなどのトリガとなるインタラクションを起こすことで、マスタプロセッサ（ＭＰ）１１を効果的に使用することができる。

【００４２】

なお、本実施形態の仕組みを用いることにより、マスタプロセッサ（ＭＰ）１

１上のプログラムがシングルスレッドで動作している場合であっても割り込み処理等の回数を減らすことが可能となり、マスタプロセッサ（ＭＰ）１１の負荷を低減することができる。

【００４３】

［実施形態２］

次に、本発明の第２実施形態として、ＤＭＡ及び各ＰＥの制御に関する具体的な手順およびローカルメモリ管理のための仕組みについて説明する。

図６には、本第２実施形態のマルチプロセッサシステムの構成が示されている。本マルチプロセッサシステムでは、ローカルメモリを管理するためのテーブルを設け、どのＶＴＩＤに対応したプログラムやデータがローカルメモリに入っているかを管理する。これにより上述の第１実施形態の効果に加え、扱うデータの不整合に関する問題を解消することができる。また、ＤＭＡコントローラやプロセッサエレメントは、対応するローカルメモリの管理テーブルを参照することにより、コマンドのプール中より実行可能なものを特定できるので、実行可能なコマンドから順に正しく実行することが可能となる。以下、具体的に説明する。

【００４４】

図６に示されているように、ＤＭＡコントローラ１５－１には、プロセッサエレメント（ＰＥ）１２－１のローカルメモリを管理する管理テーブルとしてＰＬＳ管理テーブル３０１とＤＬＳ管理テーブル３０２が設けられている。

【００４５】

ＰＬＳ管理テーブル３０１はプログラムローカルメモリ（ＰＬＳ）１３－１の記憶領域毎にどのスレッドに対応するプログラムが格納されているかを管理し、またＤＬＳ管理テーブル３０２はデータローカルメモリ（ＤＬＳ）１４－１の記憶領域毎にどのスレッドに対応するデータが格納されているかを管理する。

【００４６】

また、ＤＭＡコントローラ１５－１に設けられたＰＬＳコマンドテーブル３０３およびＤＬＳコマンドテーブル３０４は図１のコマンドプーリングバッファ１５１に相当するものであり、ＰＬＳコマンドテーブル３０３にはプログラムローカルメモリ（ＰＬＳ）１３－１とグローバルメモリ（ＧＭ）１７との間のＤＭＡ

転送に関するコマンドが蓄積され、またＤＬＳコマンドテーブル３０４にはデータローカルメモリ（ＤＬＳ）１４－１とグローバルメモリ（ＧＭ）１７との間のＤＭＡ転送に関するコマンドが蓄積される。

【００４７】

プロセッサエレメント（ＰＥ）１２－１に設けられたＰＥコマンドテーブル３０５は、図１のコマンドプーリングバッファ１２１に相当し、ここにはプロセッサエレメント（ＰＥ）１２－１に対するコマンドが蓄積される。

【００４８】

同様に、ＤＭＡコントローラ１５－２にも、プロセッサエレメント（ＰＥ）１２－２のローカルメモリを管理する管理テーブルとしてＰＬＳ管理テーブル４０１とＤＬＳ管理テーブル４０２が設けられている。ＰＬＳ管理テーブル４０１はプログラムローカルメモリ（ＰＬＳ）１３－２の記憶領域毎にどのスレッドに対応するプログラムが格納されているかを管理し、またＤＬＳ管理テーブル４０２はデータローカルメモリ（ＤＬＳ）１４－２の記憶領域毎にどのスレッドに対応するデータが格納されているかを管理する。また、ＤＭＡコントローラ１５－２に設けられたＰＬＳコマンドテーブル４０３およびＤＬＳコマンドテーブル４０４は図１のコマンドプーリングバッファ１５１に相当するものであり、ＰＬＳコマンドテーブル４０３にはプログラムローカルメモリ（ＰＬＳ）１３－２とグローバルメモリ（ＧＭ）１７との間のＤＭＡ転送に関するコマンドが蓄積され、またＤＬＳコマンドテーブル４０４にはデータローカルメモリ（ＤＬＳ）１４－２とグローバルメモリ（ＧＭ）１７との間のＤＭＡ転送に関するコマンドが蓄積される。プロセッサエレメント（ＰＥ）１２－２に設けられたＰＥコマンドテーブル４０５は、図１のコマンドプーリングバッファ１２１に相当し、ここにはプロセッサエレメント（ＰＥ）１２－２に対するコマンドが蓄積される。

【００４９】

なお、ＤＭＡコントローラは必ずしも２つ設ける必要はなく、図７に示すように一個のＤＭＡコントローラ１５のみをバス１０に接続し、そのＤＭＡコントローラ１５によってプロセッサエレメント（ＰＥ）１２－１，１２－２それぞれのローカルメモリとグローバルメモリ（ＧＭ）１７との間のＤＭＡ転送を行うよう

にしても良い。これは第１実施形態についても同様である。

【００５０】

また、一個のＤＭＡコントローラ１５のみを使用する場合には、プロセッサエレメント（ＰＥ）１２－１側に対応するＰＬＳ管理テーブル３０１、ＤＬＳ管理テーブル３０２、ＰＬＳコマンドテーブル３０３、ＤＬＳコマンドテーブル３０４と、プロセッサエレメント（ＰＥ）１２－２側に対応するＰＬＳ管理テーブル４０１、ＤＬＳ管理テーブル４０２、ＰＬＳコマンドテーブル４０３、ＤＬＳコマンドテーブル４０４が全てＤＭＡコントローラ１５に設けられることになる。また、ＰＥコマンドテーブル３０５，４０５の実際の実装位置についても必ずしもプロセッサエレメント（ＰＥ）１２－１，１２－２にする必要はなく、ＤＭＡコントローラ１５に設けても良い。第１実施形態についても同様である。

【００５１】

次に、各テーブルの具体的な構成について説明する。

本第２実施形態では、複数のスレッドが同じプロセッサエレメント（ＰＥ）およびそのローカルメモリＰＬＳ，ＤＬＳを使用しながら並列動作することを前提としている。その為に、プロセッサエレメント（ＰＥ）１２－１のプログラムローカルメモリ（ＰＬＳ）１３－１およびデータローカルメモリ（ＤＬＳ）１４－１についてはそれぞれＰＬＳ管理テーブル３０１およびＤＬＳ管理テーブル３０２によってその使用領域をスレッド毎に管理し、またプロセッサエレメント（ＰＥ）１２－２のプログラムローカルメモリ（ＰＬＳ）１３－２およびデータローカルメモリ（ＤＬＳ）１４－２についてはそれぞれＰＬＳ管理テーブル４０１およびＤＬＳ管理テーブル４０２によってその使用領域をスレッド毎に管理している。

【００５２】

これらテーブルの構造はプロセッサエレメント（ＰＥ）１２－１，１２－２のどちらについても同じであるので、以下ではプロセッサエレメント（ＰＥ）１２－１および１２－２について共通に説明することにする。

【００５３】

ＰＬＳ管理テーブルとＤＬＳ管理テーブルはハードウェア簡単化のため、ペー

ジ単位で管理される。例えば、ページサイズが４ＫｂｙｔｅでＰＬＳ，ＤＬＳのサイズがそれぞれ６４Ｋｂｙｔｅであれば、それぞれ１６ページとして管理されることになる。この場合、ＰＬＳ管理テーブルはページに対応する１６のエントリを持ち、それぞれのエントリがそのページに入っているプログラムのＶＴＩＤを示すことになる。図８にＰＬＳ管理テーブルの例を示す。

【００５４】

図８では、ページ１，２，３にＶＴＩＤ４４のプログラムが入っていてその他のページは使われていない様子が示されている。

【００５５】

次に各プロセッサエレメント（ＰＥ）で処理されるデータまたは処理された結果のデータが入るＤＬＳを管理するためのＤＬＳ管理テーブルについて説明する。図９はＤＬＳ管理テーブルの例である。

【００５６】

図９（ａ）ではページ０，１はＶＴＩＤ４４のデータが入っている（Ｄは処理されるべきデータが入っていることを示す）。またページ２，３はＶＴＩＤ４４のために領域が予約されていることを示している（Ｒは予約されている領域であることを示す）。領域の予約は、対応するプロセッサエレメント（ＰＥ）がこの領域を処理結果データの書き込みのために使用することを示している。予約されたページはプロセッサエレメント（ＰＥ）がデータを書き込むとさらにＤの部分にビットが立ち図９（ｂ）のようになる。

【００５７】

次に、ＰＬＳコマンドテーブルの構成を図１０に示す。ＰＬＳコマンドテーブルはグローバルメモリ（ＧＭ）１７よりプログラムローカルメモリ（ＰＬＳ）に転送するＤＭＡのコマンドを入れておくテーブルである。マスタプロセッサ（ＭＰ）１１は、プロセッサエレメント（ＰＥ）に実行させるプログラムをグローバルメモリ（ＧＭ）１７より制御対象のプロセッサエレメント（ＰＥ）のプログラムローカルメモリ（ＰＬＳ）に転送するためにこのテーブルにＤＭＡのコマンドを登録する。

【００５８】

図１０では、ＰＬＳコマンドテーブルに二つのエントリが存在している、一つはＶＴＩＤ４４でグローバルメモリ（ＧＭ）１７のアドレス０×１２００００からプログラムローカルメモリ（ＰＬＳ）のページ０，１，２に転送するコマンド、もう一つは、ＶＴＩＤ５０でグローバルメモリ（ＧＭ）１７のアドレス０×１４００００からプログラムローカルメモリ（ＰＬＳ）のページ１４，１５に転送するコマンドである。ＰＬＳコマンドテーブルの内容はＰＬＳ管理テーブルと比較され、転送先ページが空いている場合にはそのエントリに入っているコマンドのＤＭＡが実行され、ＤＭＡが実行されるとそのエントリは消去される。複数のエントリがＤＭＡ実行可能であるときは先入れ先出しし方式で実行される。エントリの数にはハードウェア的な制約があるのでエントリがいっぱいのときはマスタプロセッサ（ＭＰ）１１のコマンド発行元のスレッドがスリープするか、エントリが空くまで待つなどの処理を行う。

【００５９】

ＰＬＳコマンドに対するＤＭＡコントローラの処理手順を図１１に示す。ＤＭＡコントローラは、まず、ＰＬＳコマンドテーブルからコマンドを取得し（ステップＳ１２１）、そのコマンドで指定される転送先ページがプログラムローカルメモリ（ＰＬＳ）上で空いているか否かをＰＬＳ管理テーブルを参照して判断する（ステップＳ１２２）。空いていない場合には、次のコマンドエントリに対する処理が行われる。一方、空いている場合には、ＤＭＡコントローラはその取得したコマンドの処理、つまりグローバルメモリ（ＧＭ）１７からプログラムローカルメモリ（ＰＬＳ）へのプログラムのＤＭＡ転送、を実行する（ステップＳ１２３）。次いで、ＤＭＡコントローラは、ＰＬＳ管理テーブルの該当エントリにＤのフラグをセットしてその内容を更新し（ステップＳ１２４）、そしてＰＬＳコマンドテーブルから該当するコマンドエントリの内容を削除した後に（ステップＳ１２５）、実行したコマンドに対応するＶＴＩＤと共にコマンド完了通知をマスタプロセッサ（ＭＰ）１１に発行する（ステップＳ１２６）。

【００６０】

このようにして、ＤＭＡコントローラは、ＰＬＳコマンドテーブルに蓄積されている複数のコマンドを、実行可能なものから順次実行する。

【００６１】

次にＤＬＳコマンドテーブルの構成を図１２に示す。ＤＬＳコマンドテーブルはグローバルメモリ（ＧＭ）１７よりデータローカルメモリ（ＤＬＳ）にデータを転送するＤＭＡのコマンドおよびデータローカルメモリ（ＤＬＳ）からグローバルメモリ（ＧＭ）１７にデータを転送するコマンドを、マスタプロセッサ（ＭＰ）１１より受け取っていれておくためのテーブルである。

【００６２】

図１２ではＤＬＳコマンドテーブルに二つのエントリがある。一つはＶＴＩＤ４４でグローバルメモリ（ＧＭ）１７のアドレス０×２００００からデータローカルメモリ（ＤＬＳ）へ転送することを示している。方向フィールドはＤＭＡ転送の方向を示す。「Ｄ」がグローバルメモリ（ＧＭ）１７からデータローカルメモリ（ＤＬＳ）への転送、「Ｇ」がデータローカルメモリ（ＤＬＳ）よりグローバルメモリ（ＧＭ）１７への転送を示している。また、転送ページのフィールドにおける「Ｄ」はデータが転送される領域を、「Ｒ」はプロセッサエレメント（ＰＥ）による処理結果データの書き込みに使用すべき予約領域を示している。例ではページ０，１にＧＭよりデータが転送され、ページ２，３は書き込みのために予約されている。

【００６３】

二つ目のエントリにはＰＥにより書き込まれた処理結果データをグローバルメモリ（ＧＭ）１７へ転送するためのコマンドが入っている。同じＶＴＩＤ４４のスレッドで扱われるのでＶＴＩＤは４４である。グローバルメモリ（ＧＭ）１７のアドレスは０×４００００で今度は転送の方向は「Ｇ」となり、データローカルメモリ（ＤＬＳ）よりグローバルメモリ（ＧＭ）１７への転送を示している。転送ページのフィールドではページ２，３が転送されることを示している。

【００６４】

ＰＬＳコマンドテーブルと同様に、エントリの数にはハードウェア的な制約があるのでエントリがいっぱいのときはマルチプロセッサ（ＭＰ）のコマンド発行元のスレッドがスリープするか、エントリが空くまで待つなどの処理を行う。ＤＬＳコマンドテーブルからＤＭＡコマンドが実行されるのは、次のような条件に

なる。

【００６５】

（１）方向フィールドがＤの時：　転送ページで示すＤおよびＲがＤＬＳ管理テーブルで空いており、かつＰＬＳ管理テーブルに同じＶＴＩＤが存在する時。つまり、ＤＭＡを実行した後、プロセッサエレメント（ＰＥ）が処理するプログラムがすでにプログラムローカルメモリ（ＰＬＳ）に入っていて、しかもデータを入れたり出したりするデータローカルメモリ（ＤＬＳ）の領域が他のスレッドとぶつからないことを保障している。

【００６６】

（２）方向フィールドがＧのとき：　転送ページで示すＤにＤＬＳ管理テーブルのＤにビットがたっており、またＶＴＩＤが一致していること。すなわちＶＴＩＤで示すスレッドのプロセッサエレメント（ＰＥ）の処理が終わってデータがすでにデータローカルメモリ（ＤＬＳ）に書き込まれてグローバルメモリ（ＧＭ）１７に転送すべきデータが揃ったことを保障している。

【００６７】

ＤＬＳコマンドテーブルは同じＶＴＩＤに対しては先入れ先出しで処理されるが、違うＶＴＩＤでの順序は保障する必要はない。

【００６８】

ＤＬＳコマンドに対するＤＭＡコントローラの処理手順を図１３に示す。ＤＭＡコントローラは、まず、ＤＬＳコマンドテーブルからコマンドを取得し（ステップＳ１３１）、そのコマンドで指定される方向フィールドをチェックする（ステップＳ１３２）。方向フィールドがＤ、つまりデータローカルメモリ（ＤＬＳ）への転送を示すコマンドであれば、ＤＭＡコントローラは、その取得したコマンドで指定されるＤ，Ｒの転送先ページがデータローカルメモリ（ＤＬＳ）上で空いているか否かをＤＬＳ管理テーブルを参照して判断する（ステップＳ１３３）。空いている場合には、さらに、取得したコマンドのＶＴＩＤと同じＶＴＩＤを持つプログラムがプログラムローカルメモリ（ＰＬＳ）に既に存在しているか否かをＰＬＳ管理テーブルを参照して判断する（ステップＳ１３４）。同じＶＴＩＤを持つプログラムがプログラムローカルメモリ（ＰＬＳ）に存在している場

合には、上述の（１）の条件が満たされているので、ＤＭＡコントローラは取得したコマンドの処理、つまりグローバルメモリ（ＧＭ）１７からデータローカルメモリ（ＤＬＳ）へのデータのＤＭＡ転送、を実行する（ステップＳ１３５）。次いで、ＤＭＡコントローラは、ＤＬＳ管理テーブルの該当エントリにＤ，Ｒのフラグをセットしてその内容を更新し（ステップＳ１３６）、そしてＤＬＳコマンドテーブルから該当するコマンドエントリの内容を削除した後に（ステップＳ１３７）、実行したコマンドに対応するＶＴＩＤと共にコマンド完了通知をマスタプロセッサ（ＭＰ）１１に発行する（ステップＳ１３８）。

【００６９】

上述の（１）の条件が満たされていない場合、つまり、取得したコマンドで指定されるＤ，Ｒの転送先ページが空いていない場合や、空いていても同じＶＴＩＤのプログラムが存在しない場合には、コマンド処理は行われず、次のコマンドエントリに対する処理が行われる。

【００７０】

一方、取得したコマンドで指定される方向フィールドがＧ、つまりグローバルメモリ（ＧＭ）１７への転送を示すコマンドであれば、ＤＭＡコントローラは、その取得したコマンドのＤの転送ページで指定されるデータローカルメモリ（ＤＬＳ）上の位置にデータが既に存在するか否かをＤＬＳ管理テーブルを参照して判断する（ステップＳ１３９）。存在する場合には、そのデータに対応するＶＴＩＤＴが取得したコマンドのＶＴＩＤＴと一致しているかどうかが判断される（ステップＳ１４０）。ＶＴＩＤが一致する場合には、上述の（２）の条件が満たされているので、ＤＭＡコントローラは取得したコマンドの処理、つまりデータローカルメモリ（ＤＬＳ）からグローバルメモリ（ＧＭ）１７へのデータのＤＭＡ転送、を実行する（ステップＳ１４１）。次いで、ＤＭＡコントローラは、ＤＬＳ管理テーブルの該当エントリのＤフラグをリセットしてその内容を更新し（ステップＳ１４２）、そしてＤＬＳコマンドテーブルから該当するコマンドエントリの内容を削除した後に（ステップＳ１４３）、実行したコマンドに対応するＶＴＩＤと共にコマンド完了通知をマスタプロセッサ（ＭＰ）１１に発行する（ステップＳ１４４）。

【００７１】

上述の（２）の条件が満たされていない場合、つまり、取得したコマンドで指定されるＤの転送ページが存在しない場合や、存在してもＶＴＩＤが異なる場合には、コマンド処理は行われず、次のコマンドエントリに対する処理が行われる。

【００７２】

このようにして、ＤＭＡコントローラは、ＤＬＳコマンドテーブルに蓄積されている複数のコマンドを、実行可能なものから順次実行する。

【００７３】

以上で述べたような構造は、ＤＭＡコントローラ内にあり、ステートマシンで管理更新されている。

【００７４】

次にＰＥコマンドテーブルの構成を図１４に示す。ＰＥコマンドテーブルはマスタプロセッサ（ＭＰ）１１からプロセッサエレメント（ＰＥ）へのコマンドを入れておくためのテーブルである。

【００７５】

図１４では１番目のエントリがＶＴＩＤ４４で開始ページは０。これはプロセッサエレメント（ＰＥ）がプログラムの実行を始めるべきプログラムローカルメモリ（ＰＬＳ）上のページがかかれている。使用ページフィールドはプログラムローカルメモリ（ＰＬＳ）上のページ毎にデータローカルメモリ（ＤＬＳ）上のどのページのデータを使うかが示されている。例では０と１ページのデータを使うことが示されている。ＰＥコマンドテーブル内のコマンドが実際に実行されるのはＰＥコマンドテーブルにある使用ページがＤＬＳ管理テーブルのＤで示されているページと一致し、且つＶＴＩＤが一致する場合である。すなわち、プロセッサエレメント（ＰＥ）で実行するデータがデータローカルメモリ（ＤＬＳ）に揃っていることを意味する。データローカルメモリ（ＤＬＳ）にプロセッサエレメント（ＰＥ）で使用するデータが揃っているということはすでにプログラムローカルメモリ（ＰＬＳ）には対応するプログラムが用意されているということなのですぐに実行が始められる。プロセッサエレメント（ＰＥ）で実行が終了した

らＰＥコマンドテーブルから対応するコマンドを消し、ＤＬＳ管理テーブルを更新する（Ｒのところにも立てる）。また、このＶＴＩＤではプログラムローカルメモリ（ＰＬＳ）に入れたプログラムを必要としないのであれば、対応するＰＬＳ管理テーブルも更新する。必要とするかどうかはプログラム中で明示的に指定するものとする。

【００７６】

ＰＥコマンドテーブルは同じＶＴＩＤに対しては先入れ先出し方式で処理されるが、違うＶＴＩＤでの順序は保障する必要はない。ＤＬＳコマンドテーブルは同じＶＴＩＤに対しては先入れ先出し方式で処理されるが、違うＶＴＩＤでの順序は保障する必要はない。

【００７７】

ＰＥコマンドに対するプロセッサエレメント（ＰＥ）の処理手順を図１５に示す。プロセッサエレメント（ＰＥ）は、まず、ＰＥコマンドテーブルからコマンドを取得し（ステップＳ１５１）、そのコマンドで指定される使用ページがデータローカルメモリ（ＤＬＳ）に存在しているか否かをＤＬＳ管理テーブルを参照して判断する（ステップＳ１５２）。存在する場合には、そのデータのＶＴＩＤが取得したコマンドのＶＴＩＤに一致するかどうかが判断される（ステップＳ１５３）。一致するならば、プロセッサエレメント（ＰＥ）は、取得したコマンドで指定される処理、つまりプログラムローカルメモリ（ＰＬＳ）のプログラムの実行処理を行う（ステップＳ１５４）。このプログラム実行により処理されたデータは該当するデータローカルメモリ（ＤＬＳ）上の位置に格納され、ＤＬＳ管理テーブルの更新が行われる（Ｒのところにも立てる）。そして、実行したコマンドに対応するＶＴＩＤと共にコマンド完了通知をマスタプロセッサ（ＭＰ）１１に発行する（ステップＳ１５５）。

【００７８】

このようにして、プロセッサエレメント（ＰＥ）は、ＰＥコマンドテーブルに蓄積されている複数のコマンドを、実行可能なものから順次実行する。

【００７９】

図１６には、ＤＭＡコントローラおよびプロセッサエレメント（ＰＥ）と各テ

ーブルとの関係が示されている。ここでは、プロセッサエレメント（ＰＥ）１２－１側に着目して説明するが、プロセッサエレメント（ＰＥ）１２－２側についても同様である。

【００８０】

図１６で示すように、マスタプロセッサ（ＭＰ）よりＰＬＳコマンドテーブル３０３、ＤＬＳコマンドテーブル３０４、ＰＥコマンドテーブル３０５にコマンドが登録され、ＤＭＡコントローラ１５－１およびプロセッサエレメント（ＰＥ）１２－１によるコマンドの実行に応じてＰＬＳ管理テーブル３０１およびＤＬＳ管理テーブル３０２が更新される。ＰＬＳ管理テーブル３０１およびＤＬＳ管理テーブル３０２がローカルメモリの管理を適切に行っているため、実行中のデータが他のスレッドにより書き換えられたりすることはないように制御される。

【００８１】

また、プロセッサエレメント（ＰＥ）１２－１に関しては、処理すべきデータが存在する時だけ対応するプログラムが動作するようになっている。すなわち、本発明によってＤＭＡとプロセッサエレメント（ＰＥ）の処理の依存関係が適切に表現されており、マスタプロセッサ（ＭＰ）の介在無しに、ＤＭＡとプロセッサエレメント（ＰＥ）を正しい順序関係で動作させることができ、マスタプロセッサ（ＭＰ）への負荷を減らすことができる。さらに、複数のスレッドが資源を共有することによって、例えばＤＭＡ転送によって処理データをローカルメモリに用意している間にプロセッサエレメント（ＰＥ）を他のスレッドによるプログラム処理で使うなどの効率的な動作が可能になる。この方法は、ローカルメモリをダブルバッファの構造にし、一つのスレッドに片方のバッファの処理をもう一つのスレッドにもう一方のバッファの処理を行わせることで特に有効になる。

【００８２】

図１７には、本マルチプロセッサシステム全体の動作の様子が模式的に示されている。

【００８３】

マスタプロセッサ（ＭＰ）上で実行されるあるスレッドは、まず、そのスレッドのＶＴＩＤ（ここではＶＴＩＤ＝１）と一緒にプロセッサエレメントＰＥ＃１

に対応するＰＥテーブルにコマンド（ＰＥコマンド）を登録する。次いで、そのスレッドは、プロセッサエレメントＰＥ＃１に実行させるべきプログラムをデータローカルメモリ（ＤＬＳ）に全て転送するのに必要な幾つかのコマンド（ＰＬＳコマンド）をＶＴＩＤ（ここではＶＴＩＤ＝１）と一緒にＰＬＳコマンドテーブルに登録する。この後、そのスレッドは、プロセッサエレメントＰＥ＃１に処理させるべきデータをデータローカルメモリ（ＤＬＳ）に全て転送するのに必要な幾つかのコマンド（ＤＬＳコマンド　方向Ｄ）をＶＴＩＤ（ここではＶＴＩＤ＝１）と一緒にＤＬＳコマンドテーブルに登録し、また処理結果データをグローバルメモリ（ＧＭ）に全て転送させるために必要な幾つかのコマンド（ＤＬＳコマンド　方向Ｇ）をＶＴＩＤ（ここではＶＴＩＤ＝１）と一緒にＤＬＳコマンドテーブルに登録する。これで、スレッドはコマンド処理待ちとなり、スリープ状態に移行し、他のスレッドがマスタプロセッサ（ＭＰ）上で実行される。もちろん、一連のコマンド登録が途中できなくなった場合には、その時点でスレッドの切り換えを行っても良い。

【００８４】

ＤＭＡコントローラ＃１によりプログラムローカルメモリ（ＰＬＳ）へのプログラムのＤＭＡ転送、およびデータローカルメモリ（ＤＬＳ）への処理対象データのＤＭＡ転送が行われると、プロセッサエレメントＰＥ＃１によるＰＥコマンドの処理が可能となり、ＰＬＳ上のプログラムがプロセッサエレメントＰＥ＃１によって実行され、処理結果データがＤＬＳに書き込まれる。これにより、ＤＭＡコントローラ＃１によるコマンド（ＤＬＳコマンド　方向Ｇ）の処理が可能となり、処理結果データがグローバルメモリにＤＭＡ転送される。この時点で、ＶＴＩＤ＝１に関する未応答のコマンドが無くなり、ＶＴＩＤ＝１のスレッドが再び起こされる。

【００８５】

以上のように、本プロセッサシステムによれば、ローカルメモリの各記憶領域毎にどのスレッドに対応するプログラムまたはデータが格納されているかを管理することにより、扱うデータの不整合に関する問題を解消することができる。さらに、ＤＭＡコントローラおよび各プロセッサエレメント（ＰＥ）は、それぞれ

対応するコマンド蓄積用バッファに蓄積されている各スレッドからのコマンドの中で、実行可能なコマンドを特定できるので、実行可能なコマンドから順に正しく実行することが可能となり、ＤＭＡコントローラおよび各プロセッサエレメント（ＰＥ）の空き時間を減らすことが可能となる。

【００８６】

なお、ここでは本システムを第１実施形態と第２実施形態に分けて説明したが、各実施形態の機能は適宜組み合わせて使用することができる。また、マスタプロセッサ（ＭＰ）によって実行されるプログラムはマルチスレッド型のものであることが最も好ましいが、並列実行可能な複数のプログラム実行単位の各々が、必要な演算処理などをプロセッサエレメントを用いて実行する構成であればよい。さらに、スレッド切り換えなどの機能はマスタプロセッサ（ＭＰ）上で実行されるオペレーティングシステムのカーネルに組み込んだり、或いは専用のプログラムを用いてスレッド切り換えの機能を実現してもよい。

【００８７】

また、本発明は、上記各実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。更に、上記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題の少なくとも１つが解決でき、発明の効果の欄で述べられている効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【００８８】

【発明の効果】

以上詳述した如く本発明によれば、マスタプロセッサの負荷の増大や、扱うデータの不整合に関する問題を招くことなく、空いているプロセッサエレメントの時間を減らしてスループットの向上を図ることが可能となる。

【図面の簡単な説明】

【図１】

本発明の第１実施形態に係るマルチプロセッサシステムの構成を示すブロック図。

【図２】

同実施形態における未応答コマンド数に応じたスレッドの状態遷移制御を説明するための図。

【図３】

同実施形態におけるマスタプロセッサの構成を示すブロック図。

【図４】

同実施形態におけるマスタプロセッサ上で実行されるスレッドの動作を示すフローチャート。

【図５】

同実施形態におけるＤＭＡコントローラおよびプロセッサエレメントの動作を示すフローチャート。

【図６】

本発明の第２実施形態に係るマルチプロセッサシステムの構成を示すブロック図。

【図７】

同実施形態のマルチプロセッサシステムの他の構成例を示すブロック図。

【図８】

同実施形態におけるＰＬＳ管理テーブルの構成を示す図。

【図９】

同実施形態におけるＤＬＳ管理テーブルの構成を示す図。

【図１０】

同実施形態におけるＰＬＳコマンドテーブルの構成を示す図。

【図１１】

同実施形態におけるＰＬＳコマンドに対するＤＭＡコントローラの処理手順を示すフローチャート。

【図１２】

同実施形態におけるＤＬＳコマンドテーブルの構成を示す図。

【図１３】

同実施形態におけるＤＬＳコマンドに対するＤＭＡコントローラの処理手順を示すフローチャート。

【図１４】

同実施形態におけるＰＥコマンドテーブルの構成を示す図。

【図１５】

同実施形態におけるＰＥコマンドに対するプロセッサエレメントの処理手順を示すフローチャート。

【図１６】

同実施形態におけるＤＭＡコントローラおよびプロセッサエレメントと各テーブルとの関係を示す図。

【図１７】

同実施形態におけるマルチプロセッサシステム全体の動作の様子を模式的に示す図。

【符号の説明】

１０…バス

１１…マスタプロセッサ（ＭＰ）

１２－１，１２－２…プロセッサエレメント（ＰＥ）

１３－１，１３－２…プログラムローカルメモリ（ＰＬＳ）

１４－１，１４－２…データローカルメモリ（ＤＬＳ）

１５－１，１５－２…ＤＭＡコントローラ

１６…メモリコントローラ

１７…グローバルメモリ（ＧＭ）

１１１…カウンタアレイ

１２１，１５１…コマンドプーリングバッファ

２０１…プロセッサモジュール

２０２…バスコントローラ

２０３…カウンタアレイ

３０１，４０１…ＰＬＳ管理テーブル

３０２，４０２…ＤＬＳ管理テーブル

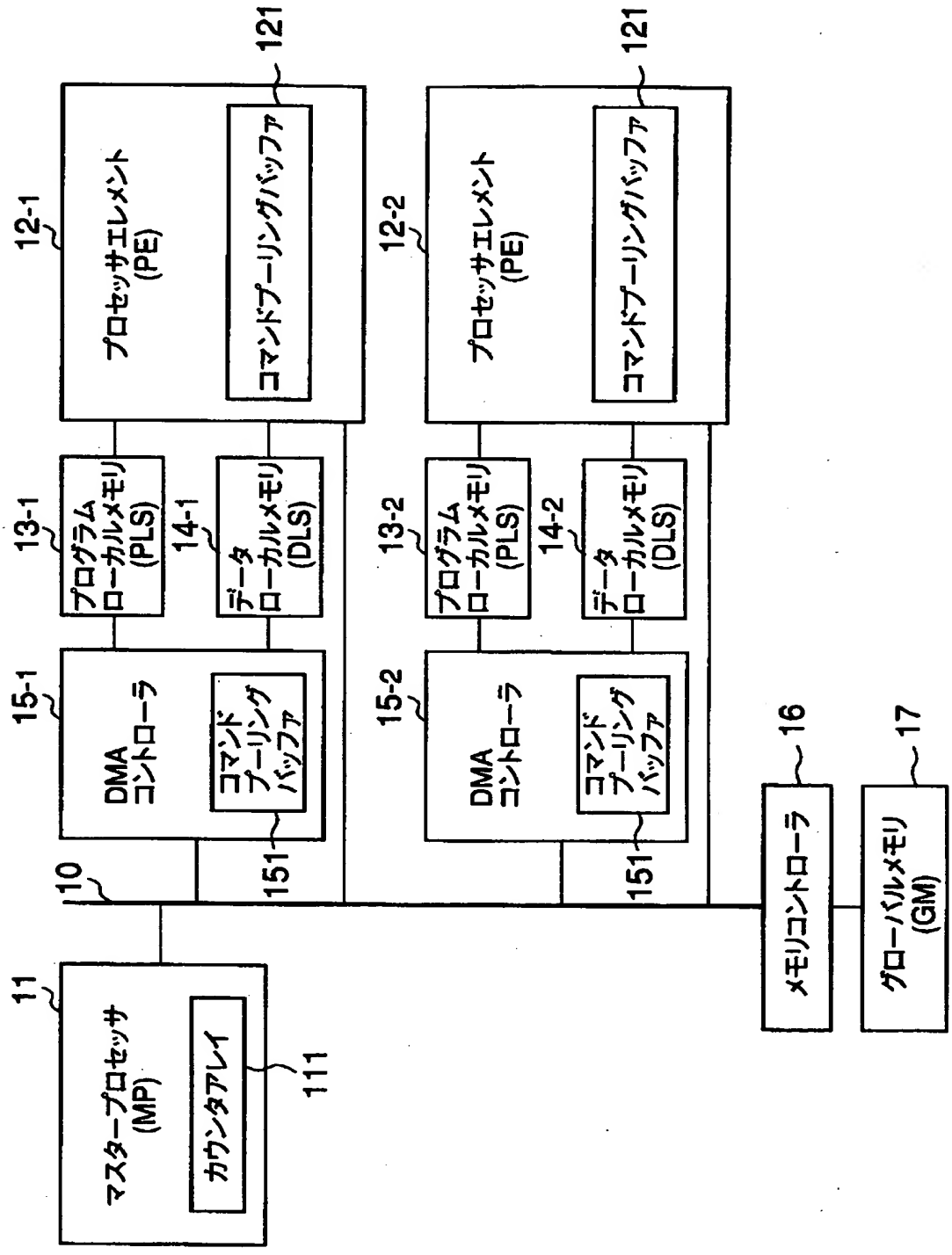
３０３，４０３…ＰＬＳコマンドテーブル

３０４，４０４…ＤＬＳコマンドテーブル

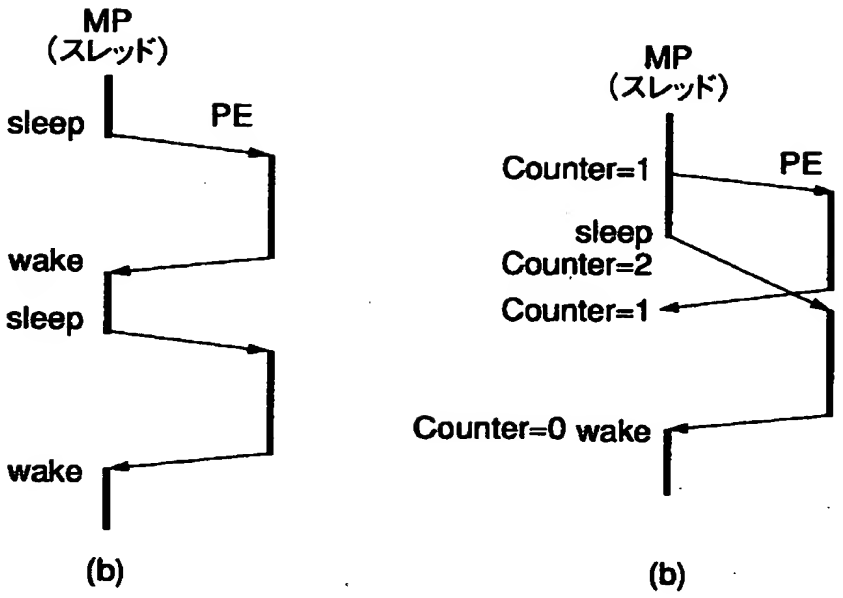
３０５，４０５…ＰＥコマンドテーブル

【書類名】　　図面

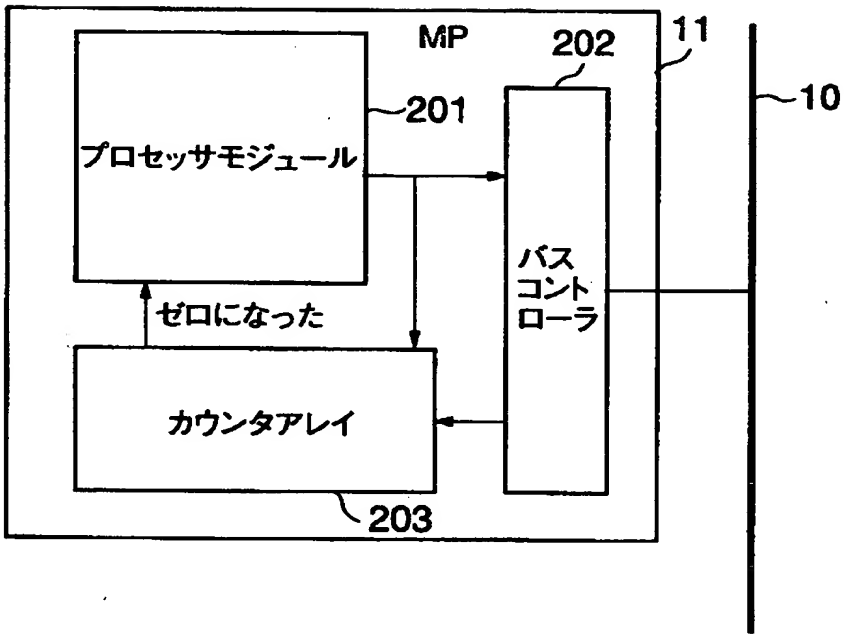
【図１】


11
マスタープロセッサ (MP)
カウンタアレイ
111
10
151
151
15-1
DMA コントローラ
コマンドブーリングバッファ
15-2
DMA コントローラ
コマンドブーリングバッファ
13-1
プログラム ローカルメモリ (PLS)
14-1
データ ローカルメモリ (DLS)
13-2
プログラム ローカルメモリ (PLS)
14-2
データ ローカルメモリ (DLS)
12-1
プロセッサエレメント (PE)
121
コマンドブーリングバッファ
12-2
プロセッサエレメント (PE)
121
コマンドブーリングバッファ
16
メモリコントローラ
17
グローバルメモリ (GM)

【図２】

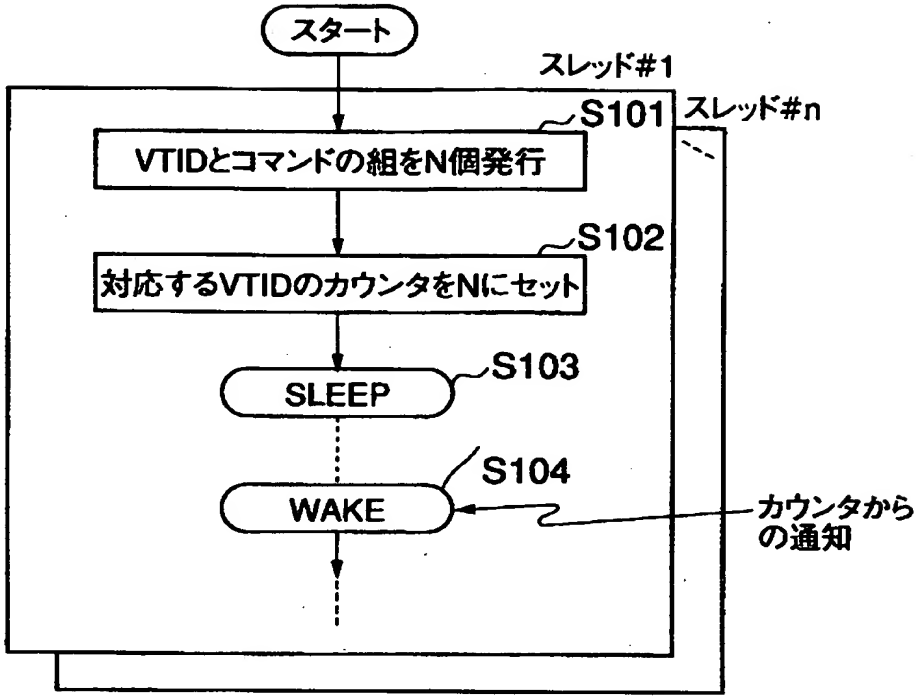


【図３】

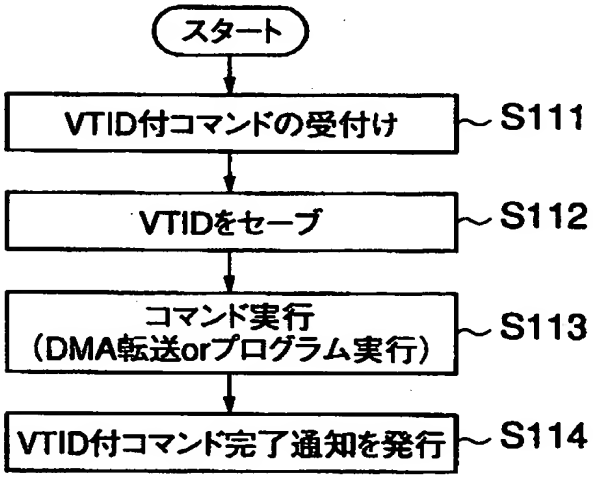


出証特２００１－３１１０９１５

【図４】

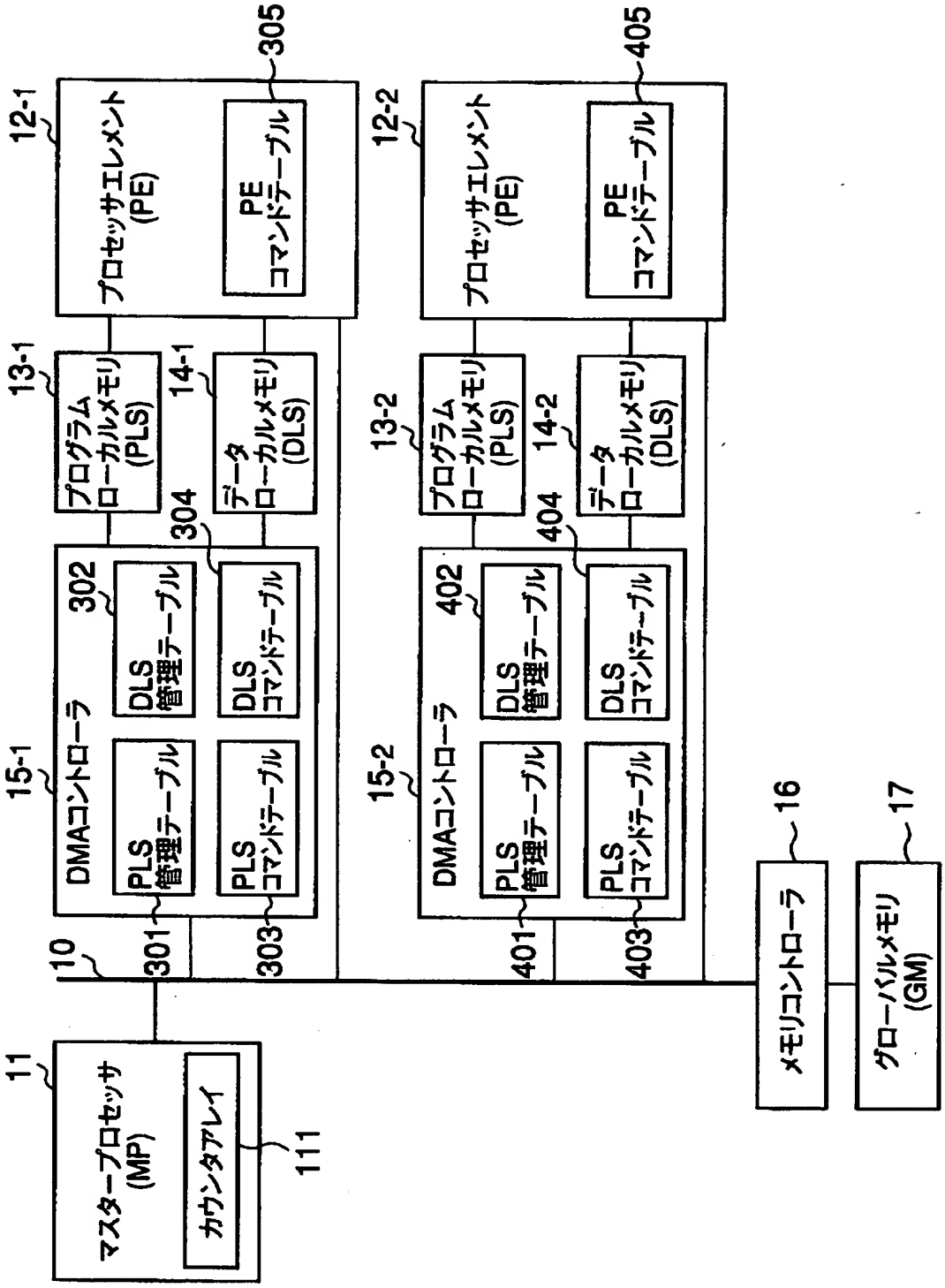


【図５】

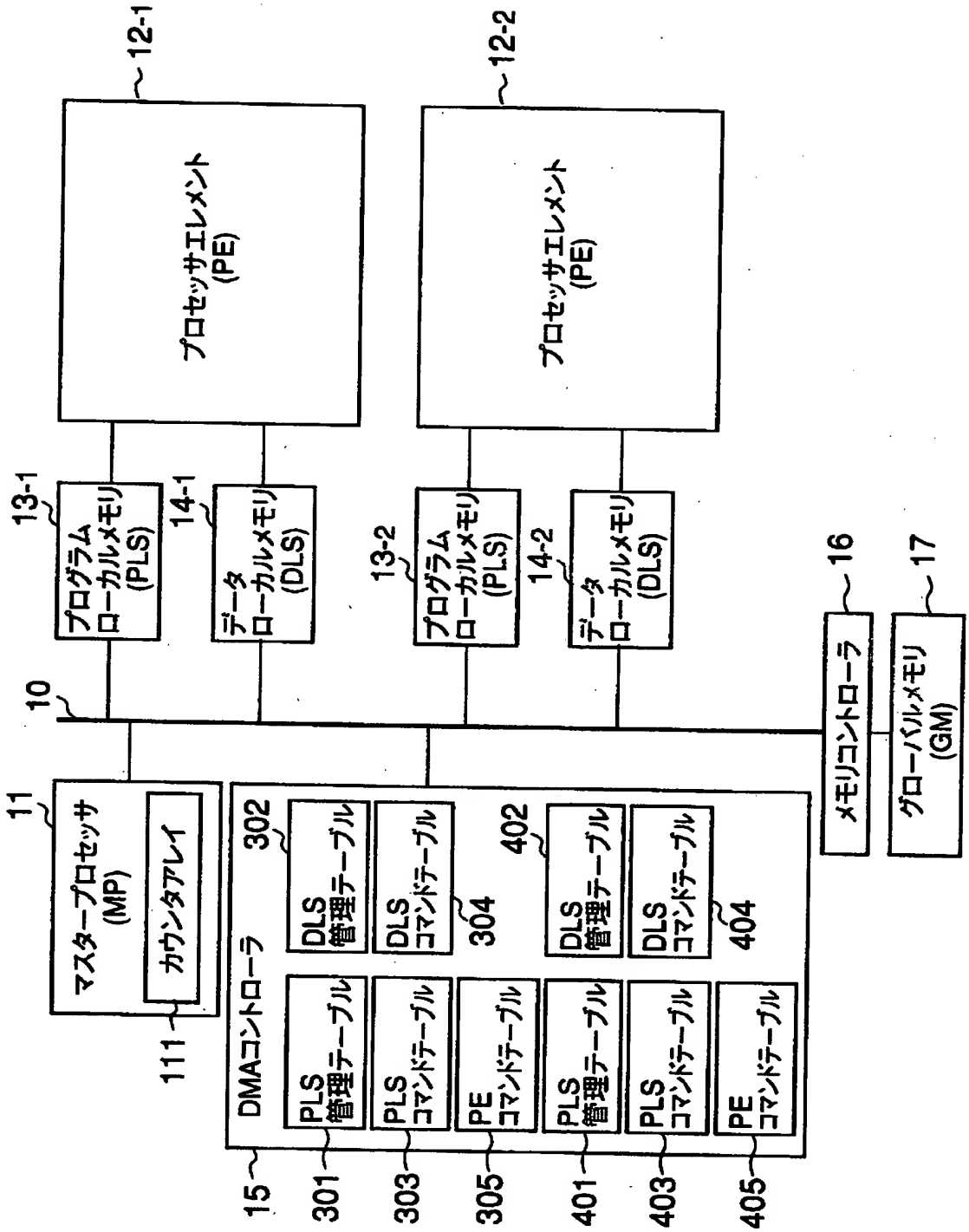


出証特２００１－３１１０９１５

【図６】


マスタープロセッサ (MP)
カウンタアレイ
11
111
10
301
303
401
403
DMAコントローラ
15-1
15-2
302
402
DLS 管理テーブル
304
404
DLS コマンドテーブル
PLS 管理テーブル
PLS コマンドテーブル
プログラム ローカルメモリ (PLS)
13-1
13-2
データ ローカルメモリ (DLS)
14-1
14-2
プロセッサエレメント (PE)
12-1
12-2
305
405
PE コマンドテーブル
メモリコントローラ
16
グローバルメモリ (GM)
17

【図７】


12-1
プロセッサエレメント (PE)
12-2
プロセッサエレメント (PE)
13-1
プログラム ローカルメモリ (PLS)
14-1
データ ローカルメモリ (DLS)
13-2
プログラム ローカルメモリ (PLS)
14-2
データ ローカルメモリ (DLS)
16
メモリコントローラ
17
グローバルメモリ (GM)
10
11
マスタープロセッサ (MP)
カウンタアレイ
111
DMAコントローラ
302
DLS 管理テーブル
DLS コマンドテーブル
304
402
DLS 管理テーブル
DLS コマンドテーブル
404
PLS 管理テーブル
PLS コマンドテーブル
PE コマンドテーブル
PLS 管理テーブル
PLS コマンドテーブル
PE コマンドテーブル
15
301
303
305
401
403
405

【図８】

PLS管理テーブル

0	44
1	44
2	44
3	—
12	—
13	—
14	—
15	—

【図９】

DLS管理テーブル

0	44	D	
1	44	D	
2	44		R
3	44		R
12	—		
13	—		
14	—		
15	—		

(a)

DLS'管理テーブル

0	44	D	
1	44	D	
2	44	D	R
3	44	D	R
12	—		
13	—		
14	—		
15	—		

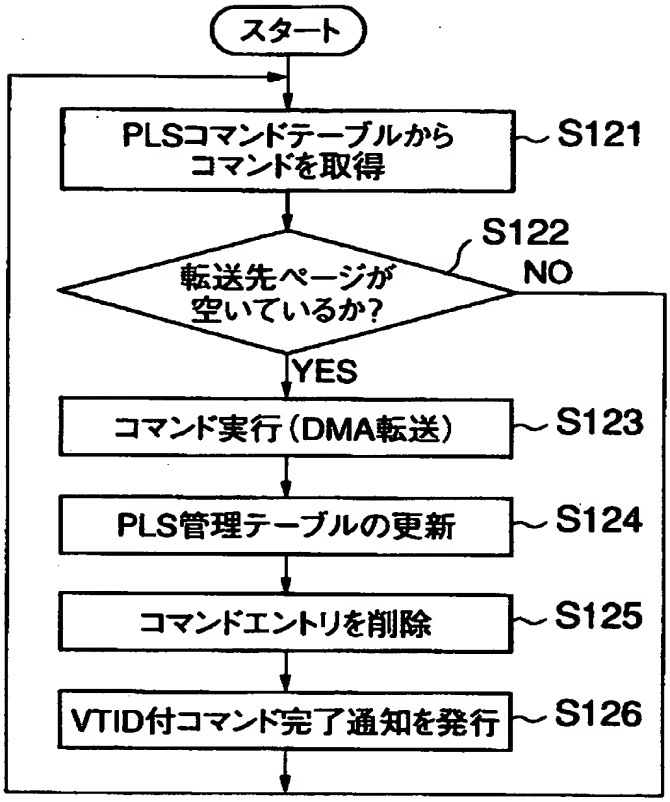
(b)

【図１０】

VTID	転送元GMアドレス	転送先ページ(ビットベクター)							
		0	1	2	3		13	14	15
44	0x120000	1	1	1	0	・・・・	0	0	0
50	0x140000	0	0	0	0	・・・・	0	1	1

PLSコマンドテーブル

【図１１】

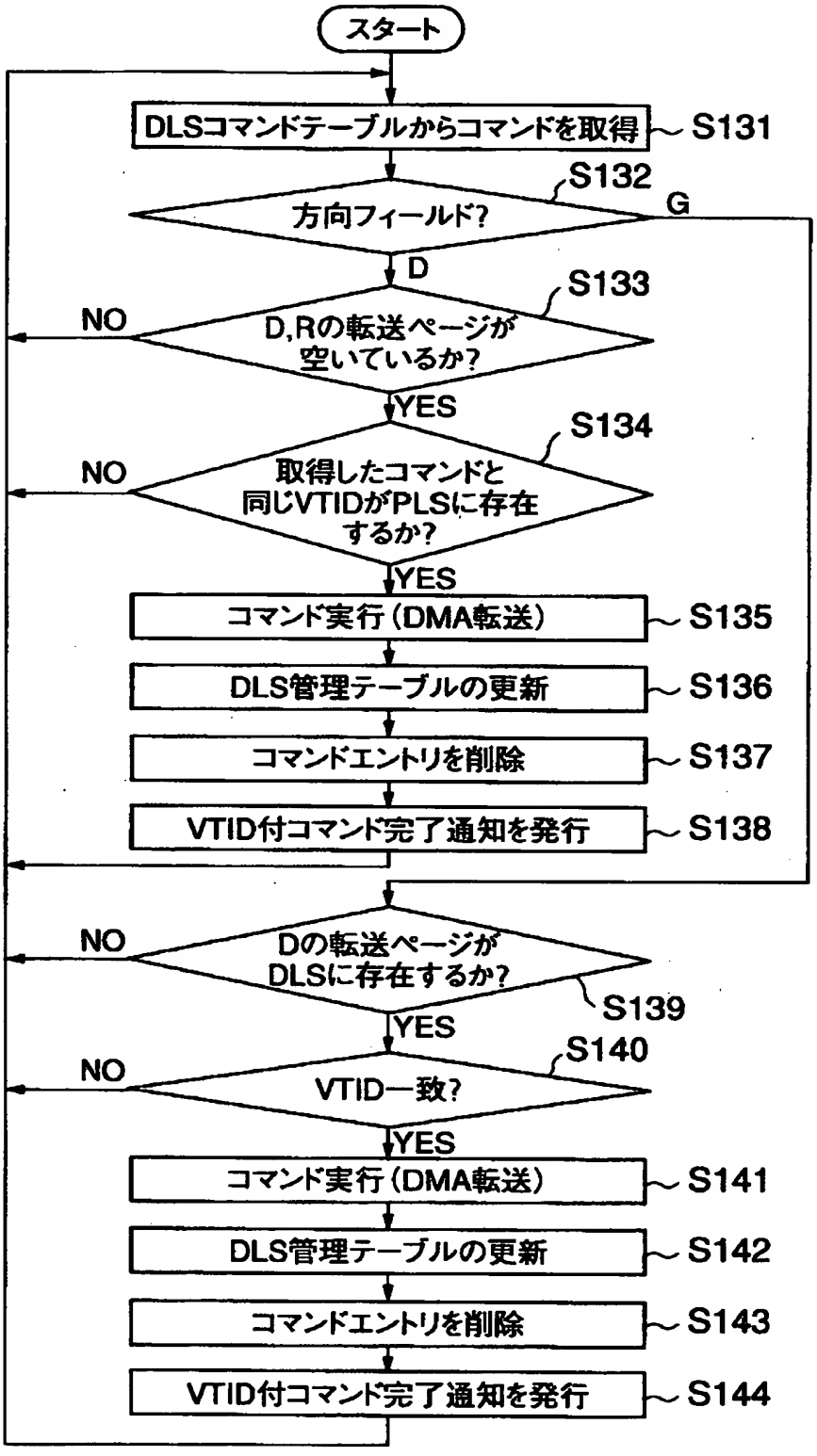


【図１２】

VTID	GMアドレス	方向	転送ページ(ページ毎)											
			0	1	2	3						13	14	15
44	0x20000	D	D	D	R	R	・	・	・	・		0	0	0
44	0x40000	G	0	0	D	D	・	・	・	・		0	0	0

DLSコマンドテーブルの構成

【図１３】

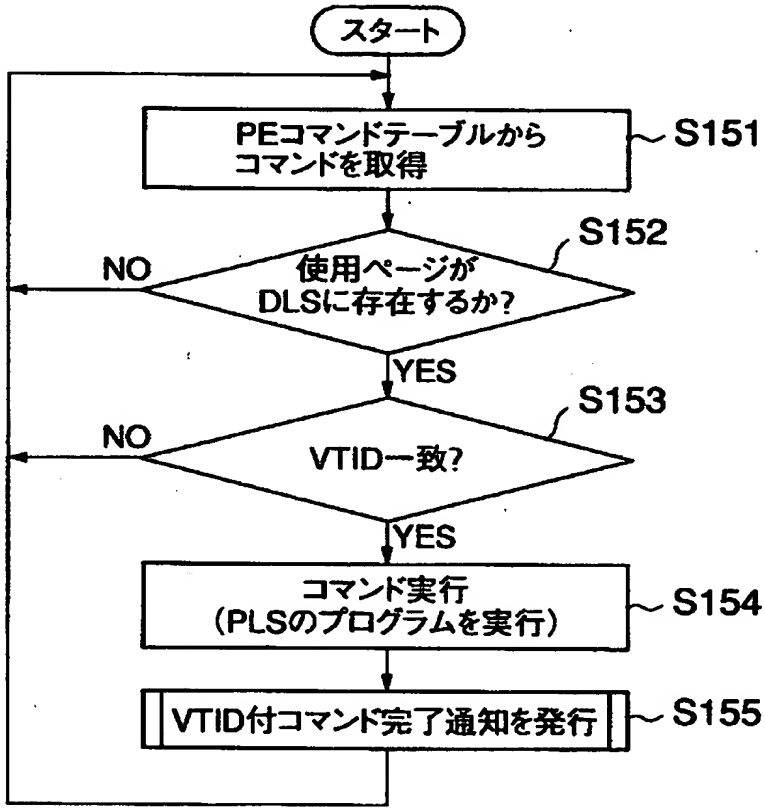

スタート
DLSコマンドテーブルからコマンドを取得
S131
S132
方向フィールド?
G
D
S133
NO
D,Rの転送ページが空いているか?
YES
S134
NO
取得したコマンドと同じVTIDがPLSに存在するか?
YES
コマンド実行(DMA転送)
S135
DLS管理テーブルの更新
S136
コマンドエントリを削除
S137
VTID付コマンド完了通知を発行
S138
NO
Dの転送ページがDLSに存在するか?
S139
YES
S140
NO
VTID一致?
YES
コマンド実行(DMA転送)
S141
DLS管理テーブルの更新
S142
コマンドエントリを削除
S143
VTID付コマンド完了通知を発行
S144

【図１４】

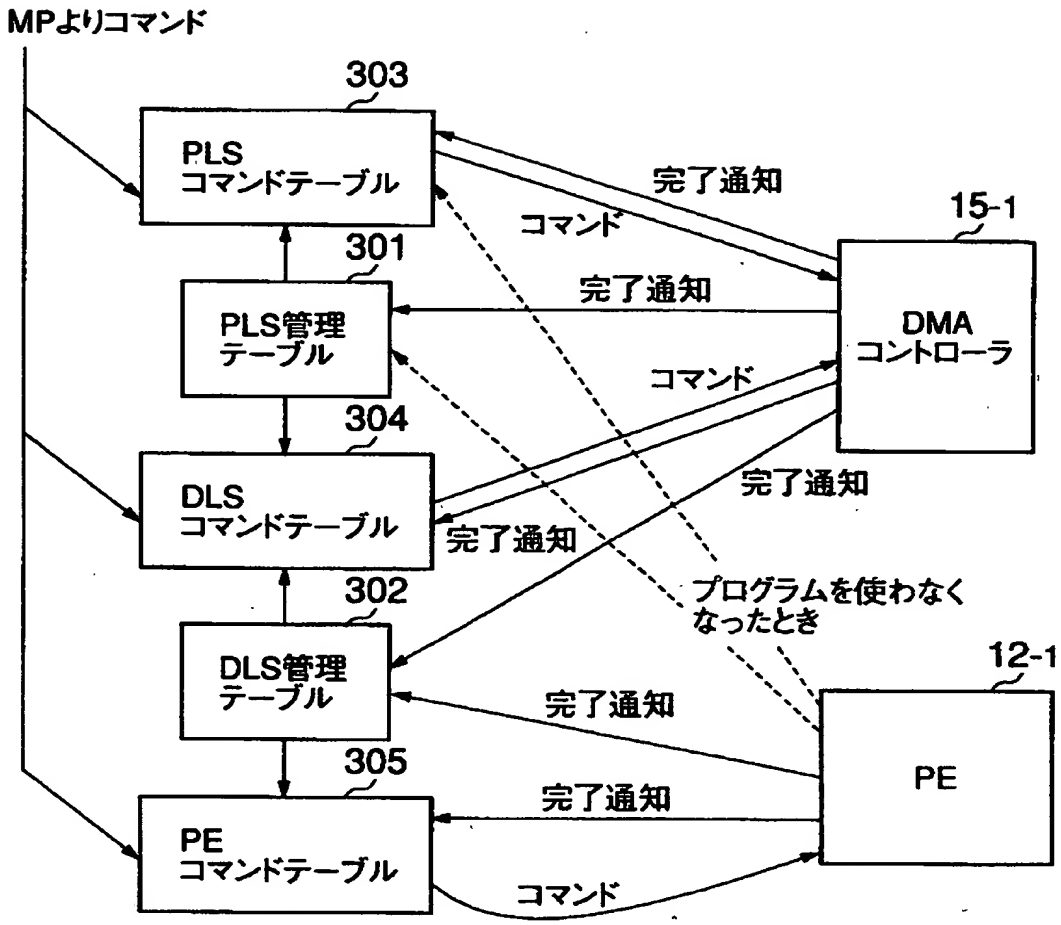
VTID	開始ページ	使用ページ(ビットベクター)								
		0	1	2	3	・・・・	13	14	15
44	0	1	1	0	0	・・・・	0	0	0
50	0	0	0	1	1	・・・・	0	0	0

PEコマンドテーブル

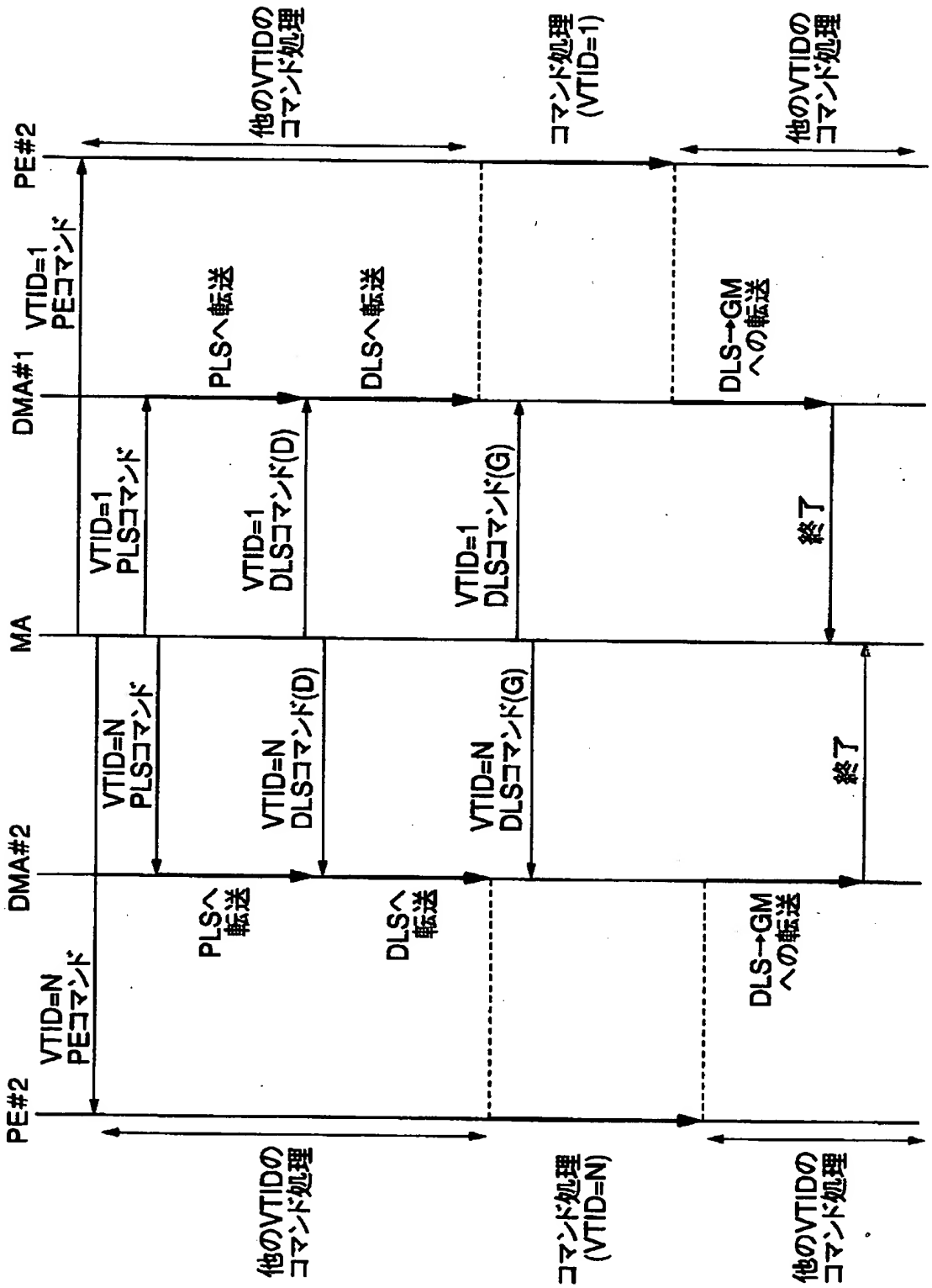
【図１５】



【図１６】


MPよりコマンド
303
PLS
コマンドテーブル
301
PLS管理
テーブル
304
DLS
コマンドテーブル
302
DLS管理
テーブル
305
PE
コマンドテーブル
完了通知
コマンド
15-1
DMA
コントローラ
プログラムを使わなく
なったとき
12-1
PE

【図１７】


PE#2
DMA#1
MA
DMA#2
PE#2
VTID=1 PEコマンド
VTID=1 PLSコマンド
VTID=1 DLSコマンド(D)
VTID=1 DLSコマンド(G)
VTID=N PEコマンド
VTID=N PLSコマンド
VTID=N DLSコマンド(D)
VTID=N DLSコマンド(G)
PLSへ転送
DLSへ転送
DLS→GMへの転送
終了
PLSへ転送
DLSへ転送
DLS→GMへの転送
終了
他のVTIDのコマンド処理
コマンド処理(VTID=1)
他のVTIDのコマンド処理
他のVTIDのコマンド処理
コマンド処理(VTID=N)
他のVTIDのコマンド処理

【書類名】　　要約書

【要約】

【課題】マスタプロセッサの負荷の増大を招くことなく、マルチプロセッサシステムのスループットの向上を図る。

【解決手段】プロセッサエレメント（ＰＥ）にはそれぞれ複数のコマンドを蓄積可能なコマンドプーリングバッファ１２１が設けられ、またＤＭＡコントローラにもそれぞれ複数のコマンドを蓄積可能なコマンドプーリングバッファ１３１が設けられている。マスタプロセッサ（ＭＰ）１１からＤＭＡコントローラや各プロセッサエレメント（ＰＥ）へのコマンドは複数まとめて発行することができ、先に送ったコマンドに対する応答を待たずに次のコマンドを発行することができる。さらに、発行済みで応答が返ってきてないコマンドの数の管理はカウンタアレイ１１１によって行われ、すべての発行済みのコマンドに対して応答が返ってきたときにそのことがマスタプロセッサ（ＭＰ）１１に通知される。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００３０７８〕

１．変更年月日　　１９９０年　８月２２日
［変更理由］　新規登録
住　所　神奈川県川崎市幸区堀川町７２番地
氏　名　株式会社東芝

２．変更年月日　　２００１年　７月　２日
［変更理由］　住所変更
住　所　東京都港区芝浦一丁目１番１号
氏　名　株式会社東芝